DOCUMENT RESUME

ED 105 847 IR 001 878

AUTHOR	Olivier, W. P.
TITLE	The Ontario CAI Network.
INSTITUTION	Ontario Inst. for Studies in Education, Toronto.
PUB DATE	Jan 75
NOTE	9p.; Paper presented at the Association for the Development of Computer-Based Instructional Systems Winter Meeting (Charleston, South Carolina, January 28-30, 1975)
EDRS PRICE	MF-$0.76 HC-$1.58 PLUS POSTAGE
DESCRIPTORS	*Computer Assisted Instruction; Computer Oriented Programs; Computers; Curriculum Development; Higher Education; Man Machine Systems, *Networks; *Program Descriptions; Programed Materials; *Regional Programs; Research and Development Centers; Time Sharing
IDENTIFIERS	*Ontario; Ontario CAI Network; Ontario Institute for Studies in Education

ABSTRACT

The evolution and current operation of the Ontario Computer Assisted Instruction (CAI) Network are described. Sponsored by the Ontario Institute for Studies in Education and including 11 community colleges in Ontario, the network has computer installations and access devices throughout the province. Initial development work was done using a Digital Equipment Corporation (DEC) System 10 computer operated by the National Research Council. More recently, a Burroughs B-6700 has been brought into the network. Courseware development has taken place on the campuses of participating institutions and the primary programing languages have been FORTRAN, ALGOL, and CAN which is an intermediate level system control language for CAI. (DGC)

THE ONTARIO CAI NETWORK

W. P. Olivier
The Ontario Institute for Studies in Education

The term "network" seems to be in vogue now and is used to describe any collection of computer hardware spread over a large geographical area. This paper will discuss, in a non-technical manner, how the network concept is used and implemented by the Individualization Project at the Ontario Institute for Studies in Education (OISE). First, it is appropriate to explain the purpose of the OISE and our involvement in CAI.

The OISE is a provincially chartered educational research & development (R&D) institute operating on a grant from the province of Ontario. A number of projects are supported, as well, through direct grants and contracts from external funding agencies. The OISE also has a large number of field centres across the province. The Institute is divided into ten academic departments having about 600 employees, 180 of which are professors. There are about 500 full-time students with a total student body of about 2000. In this academic capacity the OISE is also the graduate school of education of the University of Toronto. The Department of Computer Applications is one of the academic departments of the Institute which has students working on M.Ed., M.A. and Ph.D. degrees and professional staff working on R&D projects such as the Individualization Project. Allocation of funds to projects like the Individualization Project is made by a number of review committees composed of the OISE and external staff.

As anyone involved in the field of CAI is aware, the development of 1) student terminal hardware, 2) CAI authoring languages, 3) operating system software, 4) courseware, 5) teacher-user and -developer training and 6) dissemination of products such as that undertaken by the Indivdualization Project is very costly. Therefore, a cooperative project was the only way to support the financial and manpower requirements when no single governmental body had the mandate or economic means to support a large scale CAI developmental effort. The payoff from this cooperative venture has been faster development, acceptance and diffusion of CAI than would have been possible had cooperation not been required.

The main design goals of the project are:

1) To produce course materials which meet real needs across as many educational institutions as possible without diluting the courseware effectiveness and efficiency. Broad needs are met by program modularity and provision of optional full

program, teacher selected or learner control of courseware operation.

2) To move swiftly and without undue disruption of service from R&D to a production mode of operation. The transition from R&D computer useage, which is free, to the production phase with users paying for the service, requires a system which is economically feasible.

3) To emphasize dissemination through communications terminals, independent CAI delivery systems and interconnected CAI systems.

The participants in this cooperative development are: The National Research Council (NRC) of Canada and eleven community colleges in Ontario.

The NRC is a scientific research agency of the federal government charged with developing and implementing scientific research programs for national benefit, in cooperation with government, industry and university sectors of the community. A major part of the NRC program is the operation of a central research facility, incorporating a medium scale time-shared computer, used by educators at remote locations in the cooperative development and evaluation of CAI systems.

OISE was the first to go on-line to the NRC facility. From one teletype in 1968, the NRC project has grown across the country, involving organizations from Montreal to Calgary.

By 1974, 13 colleges in Ontario, Quebec, and Manitoba, as well as four Ontario boards of education were utilizing the developments and most were also involved in the development of additional courseware. In Ontario, three large computer systems operate in a production CAI mode in addition to the prototype NRC system, which is used only during the R&D phase.

We use the term CAI network to mean a number of interconnected CAI processors spread over a large geographical area sharing the task of executing courseware for student terminals. A CAI processor is any computer system which executes courseware for a number of students concurrently. As of this date all of the CAI processors in the Ontario CAI network are general-purpose computer systems concurrently executing other tasks and not dedicated to CAI. Within two years, however, we hope to have one or more dedicated CAI processor interfaced to the network.

The following discussion shows the evolution of the network and should clarify our concept of CAI network.

A Prototype CAI System

Starting from an implementation on our DEC TSS/9, which was run as a single user program, we wanted to achieve three objectives with a first prototype system:

1. Provide an interpreter for the NRC DECsystem-10 which could be moved to another computer without extensive recoding.

2. Provide a system that would accept a number of CAI jobs without requiring more than one copy of the system (i.e., have some characteristics of re-entrant code), and

3. Provide a means of servicing a number of student jobs through one high speed telephone line connected to a distant line concentrator computer.

The first objective was realized by coding an interpreter for the CAN authoring language in FORTRAN with a few assembly language sub-routines.

The NRC DECsystem-10 computer has 192K (36 bits/word) main memory, three disk packs of five million word capacity each, one head-per-track (half-million word capacity) swapping disk, two 7-track tape drives and 16 asynchronous communications ports. The on-line CAN system uses 18K words when only one student is signed-on and expands to a 25K main memory usage when thirty students are signed-on. For more information on the prototype installation, see Olivier (1973a). In 1973 the prototype was moved to another DECsystem-10 at the University of Wertern Ontario, but the real test came when McGill University in Montreal decided to discontinue using their language MULE and adopt CAN.

The transportable interpreter turned out to be not as practical as hoped; because it is relatively inefficient - on the order of magnitude of twelve times from the current production system. Also, there is a large amount of work to be done to convert an interactive, multi-user FORTRAN program from a computer of one manufacture to another. The last effort was to move the McGill version to the University of Waterloo, and this was done with considerable trouble attributed to the differences in the time-sharing operating systems on the IBM 370/158 computers at McGill and Waterloo.

The second goal was achieved on the prototype system through a cooperative programming venture. Since the DECsystem-10 FORTRAN compiler would not produce re-entrant code, the OISE wrote an interpreter that appears to be working on only one student. A co-routine, OPSYS, written by NRC staff in assembly language schedules users and swaps student data and courseware areas. Another function of OPSYS is the concentration and deconcentration of student messages, and this aspect leads to the third and last objective.

The cost of data communications in Canada is relatively high, and distances are great. Any implementation of a CAI system which serves widely distributed student terminals must minimize the cost of the data communications. For example, there are terminals from Calgary, Alberta to Montreal, Quebec. We decided to adapt a portion of a DEC TSS/8 system, which we owned and was being under utilized, to the task of line concentrator computer (LCC). The LCCs task is relatively simple. While a student keys an answer, the message he is constructing is held in the LCC memory. When the student indicates he is finished by pressing a key such as the return key, the LCC schedules the student's message. As soon as a high speed line to it's CAI machine is free, the message is sent. The LCC also sends data which is used to verify that the message was transmitted accurately. In the event of a transmission error, the CAI machine requests a retransmission. In case the high speed telephone line is out, the operator's console on the LCC begins printing diagnostic messages so the operator can, through a broadcast facility, notify the students who are on-line and also begin corrective action. The LCC thus concentrates one series of key strokes into a message for transmission and reverses the situation on output. The LCC also does code translation such as convert IBM code to ASCII, schedule output to maximize throughput and act as a device controller for devices such as a slide projector. A recent study led us to recommend that Seneca College buy a LCC for $15,000 to provide 40 ports in-house. The LCC is located in the same room as their CAI machine which does not reduce high speed data communication charges, but the cost of the LCC is so inexpensive in relation to the cost of ports on the large machine, that the LCC pays for itself in less than two years.

The prototype system at the NRC was useful for an initial set of courseware development activities, but we realized that the FORTRAN based system would be too costly when the colleges acquired computing power elsewhere. The NRC participation is only for research; therefore when a courseware project was moved to production, other computing resources had to be found. We are still the largest user of the NRC computer, but our courseware projects change periodically.

The Production Systems

In the fall of 1973 we moved one course into production. The prerequisite college mathematics program, a 25-hour average, pre-calculus mathematics course, had been undergoing field trials and revisions for two years on the NRC machine and was the first course placed in production status.

Seneca College in metro Toronto had acquired a Burroughs B-6700 to use for data processing, administration and CAI; and it was our task to implement the CAN system. Two goals were uppermost in our minds -- reliability and efficiency.

The Burroughs B-6700 system has a main memory of 1100 bytes (48 bits/word), forty megabytes of head-per-track disk, 242 megabytes of disk pack and four 9-track tape drives. A data communications processor with 64 asynchronous ports is used as a front end to the system. The CAN system averages 32K words of core storage to process 70 simultaneous students, but the system expands and contracts dynamically. For more information see Dinniwell, Luts, Mikichak, Naess & Olivier (1975). ALGOL is the assembly language of the B-6700; therefore the CAN system was coded in ALGOL. Transportability to a machine of another manufacture was given secondary importance. The system was designed as a set of asynchrous dependent tasks. In case any segment of the CAN system fails, it automatically initiates restart procedures. If the fault condition is in one student's area and cannot be corrected, then that student is signed off. If the problem is with one courseware segment, all students and the system error message file are informed what is wrong and the students are signed off. The software tries to recover from an error several times before notifying one of the two system operators on duty at the consoles. If the entire hardware system crashes, as it does occasionally, the CAN system automatically restarts itself and checks for student responses it had intact in the PILLOW file which had not been processed before the crash. The students with incomplete responses are asked to re-enter their last response. If the data communications is lost and students are on dial-up lines, then the CAN system asks a student to reconfirm his name, not the entire sign-on, before resuming the program exactly where he left off, not at the last restart point.

Not all of the features can be mentioned, but enough have been listed to give a flavour of the priority given to reliability and fault recovery. We feel that a student should be interrupted as little as possible in the event of system problems. Fortunately, crashes due to the CAN system are very rare, but operating system and other applications concurrently running on the system cause crashes. System instability will probably always exist in a multi-programming environment. Eventually we will implement smaller dedicated CAI machines; but the volume of CAI used in Ontario, only about 3000 students now, is not such that colleges can presently justify a machine dedicated to CAI.

The other criterion was speed of processing and this was achieved by extensive preprocessing of the CAN source code. The courseware code is compacted, usually a 60% reduction, and the intermediate language is in a byte-string stack notation The interpreter thus emulates a stack-oriented CAN machine. The intermediate language was designed to reduce the amount of courseware code and be primitive enough to be an intermediate language for other CAI languages such as COURSEWRITER. One of the longer range goals is to produce a dedicated CAI machine which, through a micro-coded control memory will execute directly the CAN intermediate language. Another problem we are working on is building preprocessors which convert other CAI languages into the CAN intermediate language. This task is much easier than trying to convert from one high level authoring language to another.

The CAN system at Seneca was successful in meeting expectations and now provides CAI service at 76 cents per student contact hour. Any college or school in Ontario can arrange with Seneca for the provision of the service for $100 to $200/month/student terminal. The charge is based on a monthly fixed rate and varies depending upon data communications charges, whether the access is through one of the LCC's and how many terminals any site will use. The fixed rate simplifies budgeting and accounting procedures for both Seneca and the users.

In 1974, Algonquin College in Ottawa acquired a DECsystem-10 for the same purpose as Seneca--to provide CAI in addition to data processing and administrative jobs. We recoded in assembly language for the DECsystem-10 and made slight modifications to the production CAN system.

The DECsystem-10 at Algonquin has 192K words (36 bits/word) of main memory, four disk packs of ten million words each, two 9-track tape drives and 32 asynchronous ports. The CAN system uses about 40K words of main memory to run 70 concurrent students. By the fall of 1974, Algonquin College was also providing CAI service. To prevent the largest colleges in the Province from fighting for customers, we suggested a CAI service centre committee. The colleges have now standardized fees and other services so that the only difference in cost to a user college is the data communications charge.

Since The OISE is responsible for all software and courseware we could see problems in maintaining a number of similar systems. Nothing could be done about the software maintenance problem, but by keeping courseware on only one machine, the task of courseware maintenance would be simplified. We also wanted a cost-viable system which fostered cooperation among users, had fail-soft in addition to highly reliable operations and which could be easily expanded and enhanced.

The initial interconnection of the NRC and Seneca computers was done to save costs. However, the network concept proved so useful that it is now being expanded. Since the CAN system on all of our installations: NRC, Seneca College, Algonquin College, and University of Western Ontario accepts LCC input, we can make each machine look like an LCC to the other machines. In effect, we have a pseudo line concentrator (PLC) in each production system. To see how the interconnected systems function, take a hypothetical student sitting at a terminal in North Bay, Ontario. Assume that the student is taking a course in electronics which is resident on the NRC computer. The student keys a response that is stored in the local LCC and then forwarded to the Seneca machine in Toronto. The Seneca computer stores and forwards the student's input to the NRC computer in Ottawa where the processing occurs. From the time the student presses the return key until his terminal begins displaying new information, only a second or less elapses.

Currently only the Seneca College and NRC computers are interconnected; Algonquin College gets NRC service by connecting one of their LCCs with all of its associated terminals to the NRC. This present arrangement at Algonquin College is unsuitable, because all terminals on one campus have to be either on the NRC computer taking courses which are under development or else be connected to their own computer for a production course. The terminals at the other campuses cannot access the NRC computer. To solve these problems, this spring we are adding the Algonquin College computer to the network so that any of the terminals connected directly or through any of their three LCCs may execute courseware at Seneca or at the NRC.

Currently, students are taking CAI, full-semester, college level credit courses in Introductory Mathematics, Business Mathematics, Technology Mathematics, and Electronics. Courses are being developed in Accounting, Chemistry, Nursing, Metrication and English. We are also developing computer-assisted testing and CMI nursing materials. All courseware is cooperatively developed on a two-year cycle. Curriculum committees specify course content and the OISE codes and evaluates the materials. Olivier (1973b) explains this aspect in more detail.

Our future plans for the network include the possibility of adding remote job entry (RJE) facilities (i.e., card reader/punch and line printer) to the LCC design. This RJE addition would allow remotely located small colleges to acquire an LCC for RJE and CAI at a low cost. This facility would essentially put a large general-purpose computer in their school.

The most interesting planned addition is a dynamically micro-programmable mini computer capable of serving as a CAI machine for 200-300 simultaneous students. Simulations have indicated that the machine design is feasible and would cost less than $200,000, a cost of less than $1,000 per terminal for the central hardware.

REFERENCES

Dinniwell, J. D., Luts, J., Mikichak, P., Naess, L., and Olivier, W. P. A delivery system for computer-assisted instruction. Toronto: Canadian Data System, 1975, in press.

Olivier, W. P. CAN-6, An Authoring Language for CAI and its companion FORTRAN coded interpreter: PDP-10 implementation. Toronto: The OISE, 1973a.

Olivier, W. P. Computer-Assisted Mathematics Instruction for Community College Students. International Journal of Man-Machine Studies. 1973b, 5 (3), 385-395.